

II. AMENDMENTS TO THE CLAIMS

The following listing of claims replaces all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method for executing a computer application installed on a computer, said method comprising the steps of:

- (a) creating a servlet instance in a server connected to the computer on a first network;
- (b) running the application on the computer to generate dynamic data, the dynamic data being available for manipulation;
- (c) intercepting and redirecting said dynamic data to a network publishing component on the computer, the network publishing component using any design pattern;
- (d) transmitting dynamic data from the network publishing component to the servlet instance; and
- (e) creating data objects and populating the data objects with the dynamic data in the server,

wherein the data objects are configured to be used by other applications on the first network.

2. (Original) The method of claim 1, further comprising the steps of:

- (a) requesting the application from a client connected to a server over a second network;

- (b) updating at least one network page with the dynamic data; and
- (c) transmitting the updated network pages to the client.

3. (Original) The method of claim 1, wherein the first network is the Internet.

4. (Original) The method of claim 2, wherein the second network is the Internet.

5. (Previously Presented) The method of claim 1, wherein the first network is selected from the group consisting of: an internal network, an Intranet, a LAN, a WAN, an internal bus, and a wireless network.

6. (Previously Presented) The method of claim 2, wherein the second network is selected from the group consisting of: an internal network, an Intranet, a LAN, a WAN, an internal bus, and a wireless network.

7. (Original) The method of claim 2, further comprising:

- (a) converting the display files of the application to network pages capable of displaying dynamic data.

8. (Previously Presented) The method of claim 7, wherein the network pages are based on extensible mark-up language (XML).

9. (Previously Presented) The method of claim 8, wherein the XML language is hypertext mark-up language (HTML).

10. (Previously Presented) The method of claim 8, wherein the XML language is wireless mark-up language (WML).

11. (Original) The method of claim 7, wherein the network pages are JavaServerPages.

12. (Original) The method of claim 2, wherein the network pages are stored on the server.

13. (Original) The method of claim 1, further comprising:

(a) creating an I/O buffer for the dynamic data in the computer.

14. (Original) The method of claim 1, wherein the computer contains the server.

15. (Original) The method of claim 1, further comprising:

(a) creating a first endpoint connection between the servlet instance and the network publishing component.

16. (Original) The method of claim 15, wherein said endpoint connection is a socket.

17. (Original) The method of claim 15, wherein said endpoint connection is a data queue object.

18. (Original) The method of claim 15, wherein said endpoint connection is a message queue.

19. (Currently Amended) A program product for use in a computer network for executing an application stored on a computer from a client, said computer program product comprising a signal-bearing medium carrying thereon:

- (a) an application invoker to start and run an application in its native environment on the computer from the client;
- (b) a plurality of network user interface pages to display the application's input/output data on the client, the network user interface pages using any design pattern;
- (c) a data redirector to redirect the application's input/output data to network user-interface pages, the input/output data being available for manipulation;
- (d) a plurality of data objects corresponding to the network user interface pages to receive the application's input/output data;
- (e) a servlet instance to dynamically update the network user interface pages with the application's input/output data; and
- (f) a network user agent to display the updated network user-interface pages on the client,

wherein the data objects are configured to be used by other applications on the computer network.

20. (Original) The program product of claim 19, further comprising a screen definition converter to convert the input/output screen definitions of the application to the network user-interface pages.

21. (Currently Amended) A computer system for executing an application, comprising:

- (a) a central processing unit;
- (b) a main memory connected to the central processing unit with a communication bus;
- (c) a data storage unit connected to a data storage interface which is connected to said communication bus;
- (d) at least one input/output device connected to said communication bus and connected to a network interface to an external computer network,
- (e) an application stored in said main memory and capable of executing on said central processing unit;
- (f) a network publishing component using any design pattern;
- (g) a data redirector to redirect the application's dynamic data to the network publishing component, the dynamic data being available for manipulation; and
- (h) an I/O buffer to store the redirected dynamic data,

wherein the data objects are configured to be used by other applications on the external computer network.

22. (Currently Amended) A computer server for accessing ~~an~~ a non-modular application stored and executing on a computer, comprising:

- (a) a central processing unit;
- (b) a network interface to connect to at least one client over a network;
- (c) a servlet instance to receive a request from the at least one client to access the non-modular application and transmit the request to the computer, ~~the request being available for manipulation~~;
- (d) a server endpoint connection for transmitting and receiving real-time data to and from the computer on which the non-modular application is executing, the real-time data being in a format not suitable for display on a network; and
- (e) a plurality of data objects to be populated with the real-time data wherein the servlet receives the real-time data from the non-modular application and populates the data objects with the real-time data, and wherein the data objects are configured to be used by other applications on the network.

23. (Original) The of claim 22, further comprising:

- (a) a plurality of network display pages, each of the network display pages unique to each input/output screen definition of the application wherein the servlet updates the network display pages for transmission to the client over the network.

24. (Currently Amended) A method for executing a computer application installed on a computer, said method comprising the steps of:

- (a) converting a plurality of display files of the application to a plurality of extensible mark-up language (XML) based network pages capable of displaying the application's dynamic data;
- (b) creating a servlet instance in a server connected to the computer on a network;
- (c) requesting the application from a client connected to a server over the Internet;
- (d) running the application on the computer in its native environment;
- (e) creating an I/O buffer in the computer for the application's dynamic data;
- (f) creating an endpoint connection between the servlet instance and a network publishing component on the computer;
- (g) transmitting the dynamic data back and forth from the client to the application through the servlet instance, ~~the dynamic data being available for manipulation;~~
- (h) creating data objects and populating the data objects with the dynamic data;
- (i) updating at least one network page using the data objects, the at least one network page using any design pattern;
- (j) transmitting the updated network pages to the client over the Internet; and
- (k) transmitting network pages having responsive data from the client to server for transmission as input data to the application,
wherein the data objects are configured to be used by other applications on the network.

25. (Currently Amended) A method of interacting with a computer application, comprising:

- (a) executing a legacy computer application in its native environment;
- (b) redirecting I/O requests from and responses to the legacy computer application from a client over the Internet without introducing changes to the code of the legacy computer application, wherein a state of the legacy computer application is preserved between a session of the client and a later session of the client.

26. (New) A method for accessing a legacy computer application over the Internet, the method comprising:

converting the proprietary screen definition to a converted user interface page in a pervasive computer Internet user agent format, the converting step further comprising:

parsing a proprietary screen definition associated with the legacy computer application, wherein the proprietary screen definition is stored separately from the legacy computer application;

mechanically mapping user interface elements of the proprietary screen definition to an extensible mark-up language (XML) based language that supports user interface elements;

customizing the converted user interface page using a customize template that identifies user interface design patterns that reflect a presentation style of a user; and

validating input fields of the converted user interface page;

providing access to the legacy computer application responsive to a request from a client that is separate from a computer on which the legacy computer application resides, the client and

computer being connected by a server, the providing step further comprising:

redirecting a raw output of the legacy computer application to a network

publishing component;

reformatting the raw output for publishing, wherein the reformatted raw output is stored in JavaBean data objects that are populated by a servlet which has received data to publish;

updating dynamically the converted user interface pages using the reformatted raw output;

sending the updated converted user interface pages to the client; and

reformatting input data from the client using the network publishing component to a format of the legacy computer application; and

making the reformatted raw output in the JavaBean data objects available for access, use and manipulation by another network based application;

wherein the legacy computer application is not restructured and no code changes are made programs of the legacy computer application,

wher cin the legacy computer application is non-modular,

wherein the legacy computer application seamlessly interweaves with the network based application, and

wherein a state of the legacy computer application is automatically preserved in a native environment after a transaction on the Internet is completed and a connection to the Internet is broken.